UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
10/813,867	03/30/2004	Geraint North	1801270.00140US1	5560

23483 7590 12/26/2008

WILMERHALE/BOSTON
60 STATE STREET
BOSTON, MA 02109

EXAMINER	
KANG, INSUN	

ART UNIT	PAPER NUMBER
2193	

NOTIFICATION DATE	DELIVERY MODE
12/26/2008	ELECTRONIC

Please find below and/or attached an Office communication concerning this application or proceeding.

The time period for reply, if any, is set in the attached communication.

Notice of the Office communication was sent electronically on above-indicated "Notification Date" to the following e-mail address(es):

michael.mathewson@wilmerhale.com
teresa.carvalho@wilmerhale.com
sharon.matthews@wilmerhale.com

Office Action Summary	Application No.	Applicant(s)	
	10/813,867	NORTH, GERAINT	
	Examiner	Art Unit	
	INSUN KANG	2193	

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE *3* MONTH(S) OR THIRTY (30) DAYS, WHICHEVER IS LONGER, FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

1)☒ Responsive to communication(s) filed on *14 October 2008*.

2a)☒ This action is **FINAL**. 2b)☐ This action is non-final.

3)☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

4)☒ Claim(s) *1-67* is/are pending in the application.

4a) Of the above claim(s) _____ is/are withdrawn from consideration.

5)☐ Claim(s) _____ is/are allowed.

6)☒ Claim(s) *1-67* is/are rejected.

7)☐ Claim(s) _____ is/are objected to.

8)☐ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

9)☐ The specification is objected to by the Examiner.

10)☐ The drawing(s) filed on _____ is/are: a)☐ accepted or b)☐ objected to by the Examiner.

Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).

11)☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

12)☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

a)☐ All b)☐ Some * c)☐ None of:

1.☐ Certified copies of the priority documents have been received.

2.☐ Certified copies of the priority documents have been received in Application No. _____.

3.☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

1) ☐ Notice of References Cited (PTO-892)
2) ☐ Notice of Draftsperson's Patent Drawing Review (PTO-948)
3) ☐ Information Disclosure Statement(s) (PTO/SB/08) Paper No(s)/Mail Date _____.
4) ☐ Interview Summary (PTO-413) Paper No(s)/Mail Date. _____.
5) ☐ Notice of Informal Patent Application
6) ☐ Other: _____.

DETAILED ACTION

1. This action is in response to the amendment filed 10/14/2008.

2. Claims 1-67 are pending in the application.

Claim Rejections - 35 USC § 102

3. The following is a quotation of the appropriate paragraphs of 35 U.S.C. 102 that form the basis for the rejections under this section made in this Office action:

> A person shall be entitled to a patent unless –
>
> (e) the invention was described in (1) an application for patent, published under section 122(b), by another filed in the United States before the invention by the applicant for patent or (2) a patent granted on an application for patent by another filed in the United States before the invention by the applicant for patent, except that an international application filed under the treaty defined in section 351(a) shall have the effects for purposes of this subsection of an application filed in the United States only if the international application designated the United States and was published under Article 21(2) of such treaty in the English language.

4. Claims 1-4, 6-9, 11, 13-15, 17-20, 22-25, 27, 29-31, 33-36, 38-41, 43, 45-47, 49, 52-54, and 56-67 are rejected under 35 U.S.C. 102(e) as being anticipated by Babaian et al. (US Patent 6,820,255) hereafter Babaian.

Per claim 1:

Babaian discloses:

- Providing a first translator instance which translates the subject code into the target code including translating a first portion of subject code into a portion of target code; (i.e. "first translate foreign binary code to equivalent host code," col. 5 lines 17-24; "where the foreign code sequence…is the code …or a portion of the code…that was previously translated," col. 9 lines 25-35)

- caching said portion of the target code (i.e. "the cache of host code," col. 3 lines 26-43; see Fig. 4 which shows the database cache for minimizing the need to translate foreign code at run-time; col. 4 lines 12-15)
- providing a second translator instance which translates the subject code into the target code including retrieving the cached portion of the target code upon a compatibility detection between said cached portion of the target code and a second portion of the subject code (i.e. "compares at least a portion of the foreign code sequence…with a portion of the foreign code stored in database…associated with the binary translated code…loads the binary translated code," col. 8 lines 37-48; col. 9 lines 23-35).

Per claim 2:

Babaian further discloses:

- wherein compatibility of cache translations and subject code to be translated is determined by cache key comparison between a cache key associated with the first portion of the subject code and the second portion of the subject code (i.e. "recognizing the foreign code uses hash coding for determining an associative location in database…to identify the location in database where corresponding translated binary code is stored," col. 10 lines 47-55; "After the hash value and entry point address into the database is determined," col. 11 lines 59-64).

Per claim 3:

Babaian further discloses:

- wherein the cache key is the byte sequence that encodes the corresponding subject code instruction sequence of the respective first and second portions of subject code (i.e. "determining an associative location in databasc," col. 10 lines 47-55).

Per claim 4:

Babaian further discloses:

- wherein the cache key is a hash of the corresponding subject code instruction sequence of the respective first and second portions of subject code (i.e. "After the hash value and entry point address into the database is determined," col. 11 lines 59-64).

Per claim 6:

Babaian further discloses:

- wherein the cache key comprises a plurality of metrics (i.e. i.e. provide it with profile information," col. 6 lines 59-67 which includes a plurality of metrics such as pc, counters, timers, call stack etc).

Per claim 7:

Babaian further discloses:

- wherein compatibility is determined by computing a cache key data structure corresponding to the subject code to be translated to a plurality of second data structures, each second data structure corresponding to a different set of cached target

code instructions (i.e. "recognizing the foreign code uses hash coding for determining an associative location in database…to identify the location in database where corresponding translated binary code is stored," col. 10 lines 47-55).

Per claim 8:

Babaian further discloses:

- including the step of executing the target code (i.e. "the binary translated image of the foreign code will be available for execution by the host processor…as the translated binary code executes," col. 6 lines 28-42).

Per claim 9:

Babaian further discloses:

- wherein translations of self-modifying code are not cached (i.e. "self-modifying code…may not be discovered at binary translation time…is saved…by the optimizing binary translation process 202," which is saved in 118 and not cached in code database 208, col. 7 lines 1-8; Fig. 2).

Per claim 11:

Babaian further discloses:

- wherein the portion of target code cached comprises one or more block translations and their respective successor lists (i.e. "If an indirect jump is detected, the process

flow proceeds back to step 302," which branches to the next instruction address locations, col. 8 lines 54-58; Fig. 3).

Per claim 13:

Babaian further discloses:

- wherein the portion of target code cached consists of a single instruction (i.e. "the corresponding host code stored in code database 208," col. 7 lines 45-53).

Per claim 14:

Babaian further discloses:

- wherein the portion of target code cached comprises all code blocks corresponding to the same starting subject address (i.e. "hot spots in the foreign code," col. 12 lines 60-67).

Per claim 15:

Babaian further discloses:

- wherein the portion of target code cached comprises a cache unit representing a discrete range of subject addresses (i.e. i.e. "using the disk sector as an address or index into the database…a block of translated binary code at the address," the address space defines a range of discrete addresses, col. 10 lines 26-40).

Per claims 17-20, 22-25, 27, and 29-31, they are the system versions of claims 1-4, 6-9, 11, and 13-15, respectively, and are rejected for the same reasons set forth in connection with the rejection of claims 1-4, 6-9, 11, and 13-15 above.

Per claims 33-36, 38-41, 43, and 45-47, they are the medium versions of claims -1-4, 6-9, 11, and 13-15, respectively, and are rejected for the same reasons set forth in connection with the rejection of claims 1-4, 6-9, 11, and 13-15 above.

Per claim 49, it is the medium version of claim 1, respectively, and is rejected for the same reasons set forth in connection with the rejection of claim 1 above.

Per claim 52:

Babaian further discloses:

- wherein the first translator instance translates the first subject program including the first portion of the subject code into the portion of the target code, and the second translator instance translates the second subject program including reusing the portion of the target code created by the first translator instance (i.e. col. 3 lines 27-38).

Per claim 53:

Babaian further discloses:

- copying the portion of target code from a private storage associated with the first translator instance to a shared code cache facility;
  retrieving the portion of target code from the shared code cache facility for reuse by the second translator instance (i.e. col. 4 lines 51-62).

Per claim 54:

Babaian further discloses:

- selectively identifying one or more static target code portions amongst a plurality of portions of the target code produced by the first translator instance, wherein the static target code portions are derived from static subject code portions in the subject code; and caching the identified static target code portions (i.e. col. 3 lines 26-43; col. 6 lines 43-50).

Per claim 56:

Babaian further discloses:

- selectively identifying, caching, identifying and discarding upon completing execution of the first translator instance (i.e. col. 3 lines 26-43; col. 6 lines 43-50) .

Per claim 57:

Babaian further discloses:

- providing a shared code cache facility to cache the portion of the target code; and selectively replacing the cached portion of target code in the shared code cache facility where the second translator instance provides an updated translation of the portion of target code (i.e. col. 3 lines 26-37; 60-67).

Per claim 58:

Babaian further discloses:

- publishing the portion of target code from the first translator instance to the shared code cache facility during execution of the first translator instance (i.e. col. 3 lines 17-25);

  retrieving the portion of the target code from the shared code cache facility for reuse by the second translator instance (i.e. col. 3 lines 26-37);

  wherein the first translator instance and the second translator instance execute concurrently (i.e. col. 5 lines 5-16).

Per claim 59:

Babaian further discloses:

- publishing at cache synchronization points having a predetermined trigger condition (i.e. col. 7 lines 54-67; col. 10 lines 41-46).

Per claim 60:

Babaian further discloses:

- wherein the predetermined trigger condition is any one or more of:

  (a) idle periods when the first translator instance is inactive;

  (b) after a threshold number of translation structures have been generated; and

  (c) upon a request by the second translator instance (i.e. col. 7 lines 54-67; col. 10 lines 41-46).

Per claim 61:

Babaian further discloses:

- providing a shared code cache facility to cache the portion of the target code; and selectively performing optimizations of the shared code cache facility (i.e. col. 11 lines 56-67).

Per claim 62:

Babaian further discloses:

wherein the optimizations comprise any one or more of:

(a) restructuring a cache directory structure of the shared code cache facility to make searches for a particular portion of target code more efficient;(b) deleting translations that have been superseded by subsequent, more optimized translations of the same subject code;(c) rearranging the shared code cache facility to locate frequently requested portions of target code near each other; (d) performing offline optimizations of cached translations; and (e) performing offline predictive translation to translate subject code which has not yet been translated by a translator instance but which a translator instance is expected to encounter (i.e. col. 10 lines 26-32; col. 11 lines 58-67).

Per claim 63:

Babaian further discloses:

- loading the portion of the target code into a shared code cache facility in a portion of memory which is shared amongst at least the first and second translator instances; copying at least one part of the shared code cache facility to a private portion of

memory associated with the second translator instance upon modification of the at least one part of the shared code cache facility by the second translator instance (i.e. col. 10 lines 26-40).

Per claim 64:

Babaian further discloses:

- providing a shared code cache facility to cache the portion of target code; and distributing the shared code cache facility amongst two or more caches (i.e. col. 11 lines 25-34).

Per claim 65:

Babaian further discloses:

- the distributing step comprises providing scoped caches, ranged caches, or cache policies (i.e. col. 10 lines 41-46).

Per claim 66:

Babaian further discloses:

- aggressively optimizing the translation in the first translator instance to provide an optimized portion of target code; and reusing the optimized portion of target code in the second translator instance (i.e. col. 7 lines 1-8; col. 11 lines 29-34).

Per claim 67:

Babaian further discloses:

- performing a less aggressive optimized translation in the second translator instance when translating the second portion of subject code for which a portion of target code is not cached (i.e. col. 11 lines 56-67).

Claim Rejections - 35 USC § 103

5. The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

> (a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negatived by the manner in which the invention was made.

6. Claims 5, 21, and 37 are rejected under 35 U.S.C. 103(a) as being unpatentable over Babaian et al. (US Patent 6,820,255) hereafter Babaian, in view of Curtis et al. (US Patent 6,826,750) hereafter Curtis, and further in view of Ronstrom (US patent 6,249,788).

Per claim 5:

Babaian discloses the cache key comprising offset and length of the subject code sequence (i.e. "the location," col. 10 lines 47-55) and (5) subject memory address where subject code instruction sequence was loaded (i.e. "address is marked," col. 10 lines 25-40).

Babaian does not explicitly disclose the key comprising a version number of the translator instances. However, Curtis teaches using a version number as a key was known in the pertinent art, at the time applicant's invention was made, to identify a specific file or a portion of the file (i.e. "The key would be the version number plus the file name," col. 8 lines 59-60). It

would have been obvious for one having ordinary skill in the art to modify Babaian's disclosed system to incorporate the teachings of Curtis by adding the translator's version number in the cache key. The modification would be obvious because one having ordinary skill in the art would be motivated to identify a file or a portion of the file faster.

Babaian and Curtis do not explicitly teach the key comprising a filename of executable and last modification time of file. However, Ronstrom teaches using a filename and timestamp as a key was known in the pertinent art, at the time applicant's invention was made, to identify a specific file (i.e. "file name, time stamps…are examples of keys stored in a B-tree," col. 1 lines 59-60). It would have been obvious for one having ordinary skill in the art to modify the system of Babaian and Curtis to incorporate the teachings of Ronstrom. The modification would be obvious because one having ordinary skill in the art would be motivated to identify a file or a portion of the file faster.

Per claim 21, it is the combination version of claim 5, respectively, and is rejected for the same reasons set forth in connection with the rejection of claim 5 above.

Per claim 37, it is the medium version of claim 5, respectively, and is rejected for the same reasons set forth in connection with the rejection of claim 5 above.

7. Claims 10, 26, and 42 are rejected under 35 U.S.C. 103(a) as being unpatentable over Babaian et al. (US Patent 6,820,255) hereafter Babaian in view of Miller ("Software Based Instruction Caching for the RAW Architecture," 5/1999).

Per claim 10:

Babaian further discloses the translation structure as a block (portion, sequence) (i.e. col. 10 lines 25-35; col. 3 lines 26-35; col. 9 lines 25-35) and dynamic binary translation (i.e. "Dynamic binary translator," col. 7 lines 9-18), which translates a simple sequence of code usually on the order of single basic block and caches the resulting sequence. Babaian does not explicitly teach that the block is indeed a basic block unit. However, Miller teaches using a basic block for a cache block was well known in the pertinent art, at the time applicant's invention was made, to ensure that "all instructions which are loaded" are executed and keep "track of entry points by keeping track of blocks (i.e. page 16, section 2.1.1 Basic Block, second paragraph)." It would have been obvious for one having ordinary skill in the art to modify Babaian's disclosed system to incorporate the teachings of Miller. The modification would be obvious because one having ordinary skill in the art would be motivated to minimize wasting of the cache space and simplify bookkeeping suggested by Miller (i.e. page 16, section 2.1.1 Basic Block, second paragraph).

Per claim 26, it is the combination version of claim 10, respectively, and is rejected for the same reasons set forth in connection with the rejection of claim 10 above.

Per claim 42, it is the medium version of claim 10, respectively, and is rejected for the same reasons set forth in connection with the rejection of claim 10 above.

8. Claims 12, 16, 28, 32, 44, and 48 are rejected under 35 U.S.C. 103(a) as being unpatentable over Babaian et al. (US Patent 6,820,255) hereafter Babaian in view of Nelson et al. (US Patent 5,475,840) hereafter Nelson.

Per claim 12:

Babaian teaches converting a protion of target code into a single cache unit comprisng a subject program (col. 8 lines 37-48; col. 9 lines 23-35). Babaian does not explicitly teach the cache comprising all its associated libaries. However, Nelson teaches caching a program and its associated libraries was known in the pertinent art, at the time applicant's invention was made, to minimize the "time delay in program start-up (i.e. col. 3 lines 15-21)." It would have been obvious for one having ordinary skill in the art to modify Babaian's disclosed system to incorporate the teachings of Nelson. The modification would be obvious because one having ordinary skill in the art would be motivated to minimize "the linking overhead (i.e. col. 2 lines 55-62)" as suggested by Nelson.

Per claim 16:

Babaian teaches the protion of target code cached comprisng a subject program (col. 8 lines 37-48; col. 9 lines 23-35). Babaian does not explicitly teach the cached portion comprising a subject libary. However, Nelson teaches caching a library was known in the pertinent art, at the time applicant's invention was made, to minimize the "time delay in program start-up (i.e. col. 3 lines 15-21)." It would have been obvious for one having ordinary skill in the art to modify Babaian's disclosed system to incorporate the teachings of Nelson. The modification

would be obvious because one having ordinary skill in the art would be motivated to minimize "the linking overhead (i.e. col. 2 lines 55-62)" as suggested by Nelson.

Per claims 28 and 32, they are the combination versions of claims 12 and 16, respectively, and are rejected for the same reasons set forth in connection with the rejection of claims 12 and 16 above.

Per claims 44 and 48, they are the medium versions of claims 12 and 16, respectively, and are rejected for the same reasons set forth in connection with the rejection of claims 12 and 16 above.

9. Claims 50, 51, and 55 are rejected under 35 U.S.C. 103(a) as being unpatentable over Babaian et al. (US Patent 6,820,255) hereafter Babaian.

Per claim 50:

Babaian discloses the first and second portion of subject code in Fig 5. Babaian does not explicitly teach that the first portion of subject code is part of a first program and the second portion of subject code is part of a second program. However, it would have been obvious for one having ordinary skill in the pertinent art to modify Babaian's disclosed system to compare the translated first portion of the first program with any other portion or code in another program (any) to see if the other program code or portion is in the database cache. The modification would be obvious because one having ordinary skill in the art would be motivated to minimize "the need to translate foreign code at run-time (col. 4 lines 10-15; col. 3 lines 35-40)."

Per claim 51:

Babaian further discloses:

- said target code is cached at the end of translation of said first program (i.e. "translate the foreign code to host code…then added to the database so that it may be subsequently accessed should the need arise," col. 3 lines 30-43).

Per claim 55:

Babaian teaches identifying one or more dynamic target code portions amongst a plurality of portions of the target code produced by the first translator instance, wherein the dynamic target code portions are derived from dynamically generated portions of the subject code (i.e.col. 6 lines 50-67). Babaian does not explicitly teach that discarding the dynamic target code portions. However, it would have been obvious for one having ordinary skill in the pertinent art to modify Babaian's disclosed system to discard the dynamic code instead of caching for cache efficiency because such dynamic data can be changed at runtime unlike static data and also for the purpose of data security.

Response to Arguments

10. Applicant's arguments filed on 10/14/2008 have been fully considered but they are not persuasive.

The applicant states that: Babaian's second translator instance is merely a software layer that checks correspondence of previously translated code, Babaian provides only one translator

instance. This one translator instance creates the translated code and stores the translated code into a code database 208. Then, later, the previously translated code is retrieved from the code database for execution by the host computer instead of performing another translation of the same sequence of foreign code. The software layer of Babaian is not capable of translating any portion of the subject code (remark, 2-3).

The applicant further states in page 3 that:

> The specification, for example at paragraphs [0157] to [0161], describes the second translator instance translating the subject code when compatibility with the cached code does not exist. The specification also describes the second translator instance retrieving the cached target code upon detecting compatibility between the cached code and a portion of subject code (i.e., when compatibility does exist with some portion of the subject code). Claim 1 recites the second instance translating subject code that is not compatible with the cached target code, while retrieving the cached target code that is compatible with the subject code.

In response, the claims do not recite that the second translator instance translates subject code that is not compatible with the cached target code while retrieving the cached target code that is compatible with the subject code. In fact, the claims recite that the second translator instance's translation includes retrieving the cached portion of the target code upon compatibility detection. The specification also describes in the second translation instance, the translator retrieves the cached code and executes without a repetitive translation if compatibility exists (i.e. 0159). As in the instant invention, Babaian eliminates repetitive retranslation in a binary translation process by reusing the previously translated code in a database cache (i.e. col. 3 lines 26-35; abstract). The subsequent translation process (a second translation instance) that correlates previously translated binary host code to foreign code is a second translation instance because the previously translated binary code was previously translated by a previous translation instance (a first translation instance) and the subsequent translation process can reuse the cached

code that was previously translated and stored in a database cache (i.e. col. 3 lines 26-35; abstract). The software layer is to check the correspondence between the foreign code and binary code stored in the database (i.e. col. 8 lines 27-47; abstract). Furthermore, Babaian clearly states that "if the host code in the database does not include corresponding translated binary code then a translation process …is invoked to translate the foreign code to host code (col. 3 lines 26-42)." Therefore, when there is no corresponding translated code (compatibility), the subsequent translation process (a second translation instance) will translate the code in the usual way. Therefore, Babaian anticipates the limitations in the claims.

Conclusion

11. THIS ACTION IS MADE FINAL. Applicant is reminded of the extension of time policy as set forth in 37 CFR 1.136(a).

A shortened statutory period for reply to this final action is set to expire THREE MONTHS from the mailing date of this action. In the event a first reply is filed within TWO MONTHS of the mailing date of this final action and the advisory action is not mailed until after the end of the THREE-MONTH shortened statutory period, then the shortened statutory period will expire on the date the advisory action is mailed, and any extension fee pursuant to 37 CFR 1.136(a) will be calculated from the mailing date of the advisory action. In no event, however, will the statutory period for reply expire later than SIX MONTHS from the mailing date of this final action.

12. Any inquiry concerning this communication or earlier communications from the examiner should be directed to INSUN KANG whose telephone number is (571)272-3724. The examiner can normally be reached on M-R 7:30-6 PM.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Lewis A. Bullock, Jr. can be reached on 571-272-3759. The fax phone number for the organization where this application or proceeding is assigned is 571-273-8300.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see http://pair-direct.uspto.gov. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free). If you would like assistance from a USPTO Customer Service Representative or access to the automated information system, call 800-786-9199 (IN USA OR CANADA) or 571-272-1000.

/Insun Kang/
Examiner, Art Unit 2193